Chapter 13

WINDOWS 8.x FACEBOOK AND TWITTER METRO APP ARTIFACTS

Swasti Bhushan Deb

Abstract The release of Windows 8.x for personal computers has increased user appetite for metro apps. Many social media metro apps are available in the Windows Store, the installation of which integrates social media platforms directly into the operating system. Metro applications enable social media platforms to be accessed without an Internet browser. The increased demand for metro apps has turned out to be a gold mine in digital forensic investigations. This is because, whenever an app is executed within an operating system, evidentiary traces of activities are left behind. Hence, it is important to locate and analyze evidentiary traces in Windows 8.x personal computer environments.

This chapter focuses on the forensic analysis of two widely-used personal computer based, social media metro apps – Facebook and Twitter. Experiments were performed to determine if the activities piloted via these metro apps could be identified and reconstructed. The results reveal that, in the case of Facebook and Twitter metro apps, potential evidence and valuable data exist and can be located and analyzed by digital forensic investigators.

Keywords: Metro apps, Windows 8.x, social media, Facebook, Twitter, artifacts

1. Introduction

Social media are driving a variety of forms of social interaction, discussion, exchange and collaboration. This makes social media the playground for cyber criminals. As enterprise networks become more secure, cyber criminals focus on exploiting social media platforms and preying on their subscribers. Social media metro apps for Windows 8.x personal computers enable users to exchange information without having to use web browsers, just like the apps on smartphones and tablets. Cy-

G. Peterson and S. Shenoi (Eds.): Advances in Digital Forensics XII, IFIP AICT 484, pp. 259–279, 2016.
DOI: 10.1007/978-3-319-46279-0_13

ber criminals can leverage metro apps to perpetrate activities such as defamation, stalking, malware distribution and catphishing.

Web browsers on personal computers have been the primary instrument for committing online fraud and criminal activities; this makes them a gold mine for digital forensic investigators. The traditional approach of accessing social media sites using a browser leaves traces such as the browsing history, cookies, caches, downloads and bookmarks, enabling forensic investigators to reconstruct user activities. As metro apps replace browsers for social media access via Windows 8.x personal computers, they will have important forensic implications. The metro environment in Windows 8.x features a tile-based start screen, where each tile represents a metro application and displays relevant information. For example, a Twitter metro app may show the latest tweets, a Facebook metro app may display the latest posts and news items and a weather app may show the current temperature and forecast. Clearly, metro apps on Windows 8.x personal computers can provide a wealth of information of use in forensic investigations.

Windows 8.x features a metro-styled interface designed for touch-screen, mouse, keyboard and pen inputs. Communications apps enable users to interact with each other via email, calendars and social networks, and other means. The Microsoft account integration feature also facilitates synchronization of user data and integration with other Microsoft services such as SkyDrive, Skype, Xbox Live, Xbox Music and Xbox Video. The immersive environment of Windows 8.x leads to each app leaving a unique set of forensic artifacts.

The Facebook metro app provides a user interface with a sidebar on the left-hand side that features messages, news feeds, events, friends, photos, groups and settings. Friend requests, the inbox and notification counters are presented at the top-right-hand corner of the app.

The Twitter metro app, which may be downloaded from the Windows Store, requires a user to login during its first use. The second launch of the app prompts the user to run it in the background. This enables the user to view quick status notifications on the lock screen. The distinctive features provided by Windows 8.x are Search and Share charms. The Search charm (keyboard shortcut WIN+Q) enables users to search Twitter for hash tags or accounts from any app. The Share charm (keyboard shortcut WIN+H) enables users to tweet from any app at any time and to share content from any app to Twitter [11].

Despite the popularity of the Facebook and Twitter metro apps in Windows 8.x personal computers, very little research has focused on evaluating their evidentiary importance. Investigating a criminal case involving social media platforms is a two-step approach: (i) obtain evidence such as login IP addresses with timestamps, registered email address and phone numbers of a suspected social media user at the service provider's end; and (ii) conduct a forensic analysis of the traces left behind by the use of browsers and apps at the user's end. This research focuses on the user's end and attempts to identify the nature and locations of the forensic artifacts that are created and retained when metro apps are utilized to access and operate Facebook and Twitter accounts.

2. Related Work

The forensic analysis of Facebook and Twitter metro apps on personal computers running Windows 8.x has largely been ignored in the literature. On the other hand, considerable research has focused on social media app forensics on Android and iOS platforms.

2.1 Windows 8.x Artifacts

Thomson [18] has researched the forensic aspects of Windows 8.x systems; in particular, the forensic artifacts specific to metro apps in the Consumer Preview 32-bit edition of Windows 8.x. Brewer et al. [3] have compared the Windows 7 and Windows 8 registries with regard to various forensic considerations; their work showed that the registry has not changed significantly from Windows 7 to Windows 8. Stormo [17] has also analyzed the Windows 8 registry and Goh [8] has discussed the challenges involved in forensic investigations of personal computers running Windows 8. Khatri's forensic blog [10] discusses the forensic importance of search histories in Windows 8.x systems. Despite these research efforts, Murphy et al. [15] state that there is a significant need to better understand the storage mechanisms and forensic artifacts related to Windows Phone 8 systems.

Iqbal et al. [9] have investigated the forensic aspects of Windows RT systems, which have some similarities with Windows 8 systems. In particular, Iqbal et al. describe the filesystem structure and potential forensic artifacts, and proceed to specify a forensically-sound acquisition method for Windows RT tablets. Lee and Chung [12] have identified and analyzed artifacts of Viber and Line in Windows 8 systems; their work is important because of the wide use of instant messaging apps.

Researchers have investigated the traces of social media activities left on computer systems. Zellers [20] has discussed keyword searches for MySpace artifacts and evidence reconstruction from a Windows personal computer. Al Mutawa et al. [1] have examined the forensic artifacts created by Facebook's instant messaging service, and describe a process for recovering and reconstructing the artifacts left on a computer hard drive. Dickson [4, 5] has developed forensic techniques for recovering artifacts from AOL Instant Messenger and Yahoo Messenger.

2.2 Social Media Artifacts

Large numbers of users connect to social media sites via mobile device apps. As a result, the forensic analysis of social media apps for mobile devices is a hot topic for research in the digital forensics community.

Al Mutawa et al. [2] have investigated the artifacts related to the use of social media apps on a variety of smartphones and operating systems. Their research has identified the locations where artifacts corresponding to Facebook, Twitter and MySpace apps are stored and how they can be recovered from the internal memory of Android and iOS smartphones. Meanwhile, Parsons [16] has developed a forensic analysis methodology for Windows 10 metro apps.

Despite the body of research related to social medial apps on mobile devices, there is limited, if any, published work on the forensic analysis of Facebook and Twitter metro apps. The goal of this research is to determine if activities piloted through these metro apps can be identified and reconstructed. The focus is on performing experimental tests and analyses of Facebook and Twitter metro app data to identify data sources of interest in forensic examinations.

3. Proposed Methodology

In a criminal investigation involving social media, the evidentiary records obtained from the Facebook Law Enforcement Portal [6] include the IP addresses corresponding to account creation, login and logout actions, along with the registered mobile number, email id, etc. corresponding to the Facebook profile/page of a probable suspect. Twitter [19] provides similar evidentiary records. However, these records are insufficient for determining whether or not a crime may have been committed using a Windows 8.x personal computer. This is because it is necessary to also seize the Windows 8.x computer allegedly used by the suspect and conduct a forensic analysis of the computer, including the metro apps. This section presents a detailed description of the forensic analysis methodology for extracting artifacts associated with Facebook and Twitter metro apps on Windows 8.x computers.

Table 1 presents the hardware and software used in the experimental methodology. Snapshots of the computer drive before and after user activities were taken using Regshot. These snapshots were used to analyze the system files and folders for changes that took place during the user activities. Regshot is an open-source registry comparison utility

Table 1. Details of tools used.

Tools	**Use**	**Version**	**License**
Laptop	Create virtual environments	Windows 8.x	Single Language
VMware Workstation	Simulate a Windows 8.x personal computer environment	11.0.0 Build-2305329	Commercial
Regshot	Monitor changes to the files and registry	1.9.0	Open Source
FTK Imager	View and analyze the virtual disks of VM1 and VM2	3.2.0.0	Free
SQLite Manager	View and analyze SQLite files	0.8.3.1	Free
WinPrefetch-View	Analyze Prefetch files	1.15	Free

that takes snapshots of the registry and system drive, compares the two snapshots and identifies the differences.

The experimental methodology involved three phases, which are described in the following sections.

3.1 Phase 1

A Windows 8.x Single Language operating system was used as the base operating system on which the test environments were installed. VMware version 11.0.0 was installed on the base operating system prior to conducting the experiments. Two Windows 8 Single Language 64-bit virtual machines (VMs) were created to simulate the test environment, one each for testing the traces of Facebook and Twitter Metro app activities. FTK Imager and SQLite Manager were also installed on the base operating systems and configured during this phase.

3.2 Phase 2

The second phase focused on preparing the test environments, VM1 and VM2. Each virtual machine was configured with two partitions – the C and D drives with 30 GB storage, 3 GB RAM, a bridge network connection and one processor (with one core). VM1 was used to conduct tests of the Facebook metro app while VM2 was used to conduct tests of the Twitter metro app. A Microsoft account was created using fictitious information on each virtual machine. The Microsoft accounts were created in order to install the metro apps.

To confirm the validity of forensic data identified during Phases 2 and 3, the two virtual machines were fresh with only the built-in metro apps; no third-party metro apps were installed prior to conducting the experiments. Regshot was the only additional software installed on the two virtual machines.

3.3 Phase 3

The following activities were performed on VM1 in chronological order:

- A snapshot of the VM1 system drive was taken using Regshot (Snapshot 1).
- The Facebook metro app was installed in VM1 from the Windows Store.
- A test subject logged into a fictitious Facebook account via the installed Facebook metro app and performed common user activities.
- A snapshot of the VM1 system drive was taken using Regshot (Snapshot 2).
- Snapshots 1 and 2 were compared using Regshot.
- The results were saved in a text file and VM1 was shut down.

The following activities were performed on VM2 in chronological order:

- A snapshot of the VM2 system drive was taken using Regshot (Snapshot 3).
- The Twitter metro app was installed in VM2 from the Windows Store.
- A test subject logged into a fictitious Twitter account via the installed Twitter metro app and performed common user activities.
- A snapshot of the VM2 system drive was taken using Regshot (Snapshot 4).
- Snapshots 3 and 4 were compared using Regshot.
- The results were saved in a text file and VM2 was shut down.

In an effort to ascertain the locations of potential forensic artifacts associated with the metro apps, Regshot was used to monitor the states of the virtual machines during every step in the experiments and the changes in the system directories were documented in detail.

Table 2. Activities performed via the Facebook and Twitter metro apps.

Metro App	**User Activities**
Facebook	The app was installed from the Windows Store and an account was created for the test subject Text messages and images were sent and received Posts were shared and tagged as like; status was updated; photos were uploaded; friends were tagged, etc.
Twitter	The app was installed from the Windows Store and an account was created for the test subject Tweets were posted; photos were uploaded along with captions Users were followed; followers were messaged Searches were made of users and hashtags from within the app

This phase also documented the system files that were created, modified or deleted by the user activities listed Table 2, as well as the locations of potential evidence left behind by user interactions with the metro apps. For each of the activities performed, a baseline snapshot of the system drive was taken using Regshot. The text file output for each comparison of snapshots was found to contain the list of system files, registry, etc., reflecting the changes that had occurred to each virtual machine as a result of user interactions. The text file outputs from Regshot were used to determine the types of files that were created, changed or deleted during the experiments as well as the locations of the potential evidence left behind.

4. Results

Windows 8.x metro apps have altered the field of digital forensics, leading to new sources of evidence while requiring forensic examiners to keep abreast of the latest app developments in order to interpret and reconstruct data of evidentiary value. From a forensic perspective, the metro user interface retains many of the key artifacts present in earlier versions of Windows, but there are several new artifacts and some previous artifacts are missing or changed. This section provides a brief overview of the app data and user data storage in Windows 8.x, followed by the experimental results.

4.1 App Data Storage and User Data Storage

According to Microsoft, app data is mutable data that is specific to an app. It includes the runtime state, user preferences and other settings. Windows 8.x manages the data store of an app, ensuring that it is isolated from other apps and other users. The contents of the data store are also preserved when a user installs an update to an app and the contents of the data store are removed cleanly when an app is uninstalled [14].

Apps manage or interact with two types of data: (i) app data, which is created and managed by an app, is specific to the internal functions and configurations of the app; (ii) user data, which is created and managed by a user when using an app, includes user-selected preferences, app configuration options, document and media files, email and communication transcripts, and database records that hold content created by the user [13].

4.2 Artifacts and Their Locations

Regshot identifies the changes that have occurred to a particular Windows system and categorizes them according to:

- **Registry Keys and Values:** Added, modified and deleted.
- **Files and Folders:** Added and deleted.
- **File Attributes:** Modified.

Analyses of the changes listed above revealed the locations and the data structures of the artifacts.

Tables 3 and 4 summarize the relevant artifacts that were created, changed or deleted, along with their locations. The text file outputs of Regshot were used to identify the locations of the relevant artifacts. All the system files, folders, registry key/values and file attributes in the Regshot outputs were explored. The metro app installation and user activities listed in Table 2 played a major role in performing this task. In order to identify the artifacts and their locations, the virtual disk of each virtual machine was imported into FTK imager as an image file and analyzed.

The `AppData\Local` directory contains data specific to each metro app and does not roam with the user. User-specific data files for the Facebook and Twitter metro apps are stored in `C:\Users\{UserName}\AppData\Local\Packages\{packageid}` where `UserName` corresponds to the Windows user name and `packageid` corresponds to the Windows Store application package identifier.

Facebook and Twitter user data reside in the packages `Facebook.Facebook_8xx8rvfyw5nnt` and `9E2F88E3.Twitter_wgeqdkkx372wm`, respectively. Figures 1 and 2 present the data structures of the Facebook and Twitter app packages located at `C:\Users\{UserName}\AppData\Local\Packages`.

Table 3. Facebook metro app artifacts.

Location	Significance
`C:\ProgramFiles\WindowsApps\Facebook.Facebook_1.4.0.9_x64__8xx8rvfyw5nnt`	Facebook app installation path
`C:\Users\{UserName}\AppData\Local\Packages\Facebook.Facebook_8xx8rvfyw5nnt\`	Path to Facebook user data, the most important artifact location
`C:\Windows\Prefetch\FACEBOOK.EXE-C042A127.pf`	Facebook app prefetch file
`C:\Users\{username}\AppData\Local\Packages\Facebook.Facebook_8xx8rvfyw5nnt\TempState`	Path to uploaded photos

Table 4. Twitter metro app artifacts.

Location	Significance
`C:\ProgramFiles\WindowsApps\9E2F88E3.Twitter_1.1.13.8_x64__wgeqdkkx372wm`	Twitter app installation path
`C:\Users\{UserName}\AppData\Local\Packages\9E2F88E3.Twitter_wgeqdkkx372wm`	Path to Twitter user data, the most important artifact location
`C:\Windows\Prefetch\TWITTER-WIN8.EXE-9C6C7EE3.pf`	Twitter app prefetch file
`C:\Users\{UserName}\AppData\Local\Packages\9E2F88E3.Twitter_wgeqdkkx372wm\TempState`	Path to uploaded photos

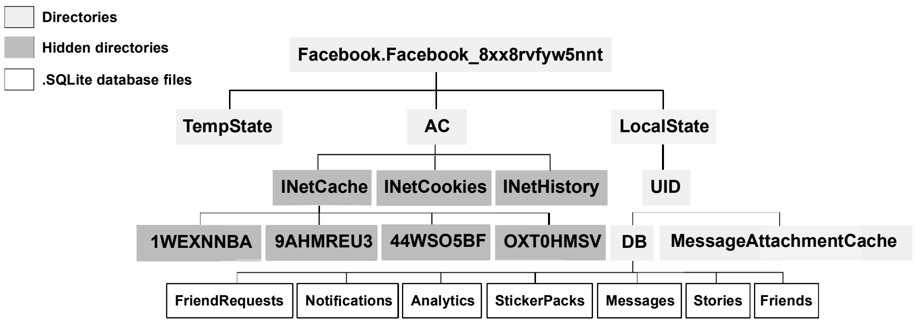


Figure 1. Data structure of the Facebook metro app package.

5. Analysis of Results

This section discusses the results of forensic analyses of the artifacts listed in Tables 3 and 4. Descriptions of the analyses and the related findings are also provided.

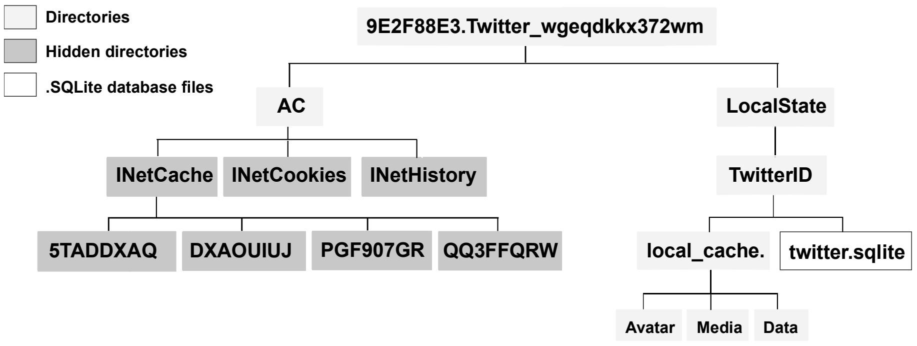


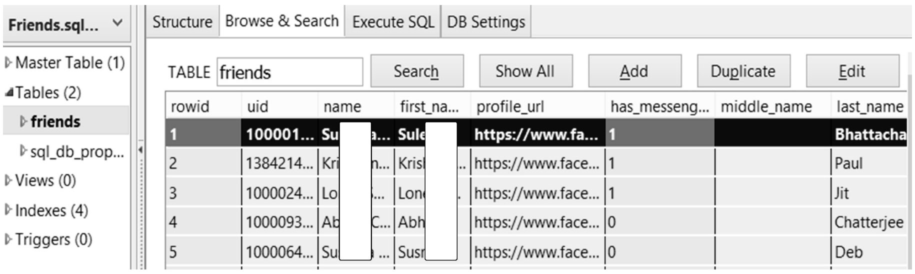
Figure 2. Data structure of the Twitter metro app package.

Table 5. Structure and forensic relevance of Facebook metro app artifacts.

Directory		Forensic Relevance
`AC`	`INetCache`	Contained four randomly-numbered hidden folders; each folder had files with the display pictures of Facebook friends, search results, etc.; each profile picture correlated with a Facebook friend (Figure 3)
	`INetCoookies`	No cookie files were found
	`INetHistory`	Found to be empty
`LocalState`	`UID/DB`	The UID corresponded to the Facebook profile id of the test subject; the directory contained database files associated with text messages, notifications, stories, friends list, etc.; this is the most important directory from the forensic perspective
	`UID/MessageAttachmentCache`	Contained randomly-numbered folders (e.g., `bef9de3f1acaaa4d34273d597809f77e.jpg`); these folders were named using the MD5 hashes of the files/pictures sent to friends via chat; the folders also contained the picture/file attachments sent to friends via chat

5.1 Analysis of Facebook Metro App Artifacts

The Facebook app identifier `Facebook.Facebook_8xx8rvfyw5nnt` located at `C:\Users\{UserName}\AppData\Local\Packages` was used to reconstruct user data, including the friends list and text messages. Table 5 shows the structure and forensic relevance of the `AC` and `LocalState` directories. Figure 3 shows the data in the "friends" table of the `Friends.sqlite` database.



Friends.sql...

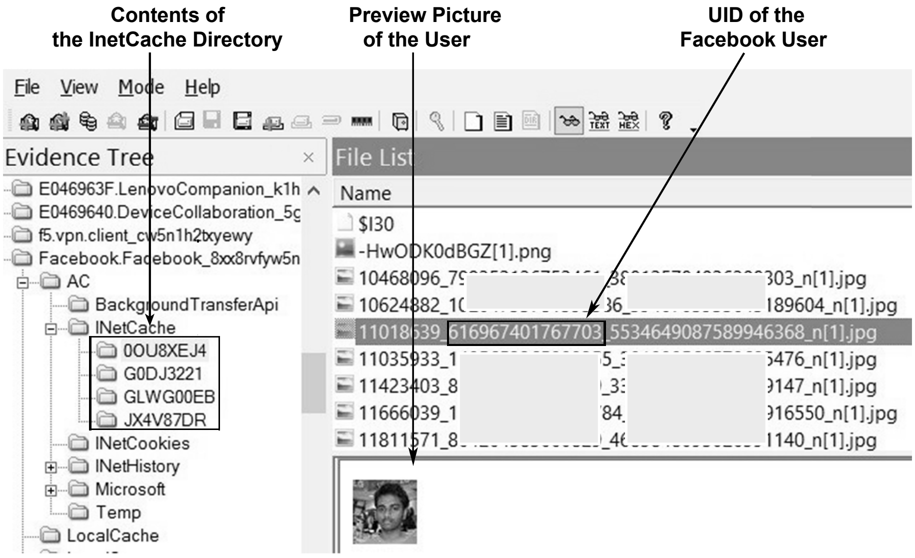
- Master Table (1)
- Tables (2)
  - **friends**
  - sql_db_prop...
- Views (0)
- Indexes (4)
- Triggers (0)

Structure | Browse & Search | Execute SQL | DB Settings

TABLE friends | Search | Show All | Add | Duplicate | Edit

rowid	uid	name	first_na...	profile_url	has_messeng...	middle_name	last_name
1	**100001...**	**Su**[redacted]**a...**	**Sule**[redacted]	**https://www.fa...**	**1**		**Bhattacha**
2	1384214...	Kri[redacted]n...	Kris[redacted]..	https://www.face...	1		Paul
3	1000024...	Lo[redacted]...	Lon[redacted].	https://www.face...	1		Jit
4	1000093...	Ab[redacted]C...	Abh[redacted]	https://www.face...	0		Chatterjee
5	1000064...	Su[redacted]...	Susr[redacted]	https://www.face...	0		Deb

Figure 3. Table viewed and examined using SQLite Manager.



- E046963F.LenovoCompanion_k1h
- E0469640.DeviceCollaboration_5g
- f5.vpn.client_cw5n1h2txyewy
- Facebook.Facebook_8xx8rvfyw5n
  - AC
    - BackgroundTransferApi
    - INetCache
      - 0OU8XEJ4
      - G0DJ3221
      - GLWG00EB
      - JX4V87DR
    - INetCookies
    - INetHistory
    - Microsoft
    - Temp
  - LocalCache

$I30
-HwODK0dBGZ[1].png
10468096_79[redacted]38[redacted]303_n[1].jpg
10624882_10[redacted]36_[redacted]189604_n[1].jpg
11018539_616967401767703_5534649087589946368_n[1].jpg
11035933_1[redacted]5_3[redacted]5476_n[1].jpg
11423403_8[redacted]_33[redacted]9147_n[1].jpg
11666039_1[redacted]84_[redacted]916550_n[1].jpg
11811571_8[redacted]_46[redacted]1140_n[1].jpg

Figure 4. Correlation of display pictures with Facebook UIDs.

The `InetCache` directory contained four randomly-numbered hidden folders. These folders contained the display pictures (`.jpg` files) of the friends/profiles of the users whose notifications appeared in news feeds, search queries, personal chats/messages, etc. The display pictures were found to correlate with the Facebook users as shown in Figure 4. FTK imager was used to view and examine the display pictures and their related information.

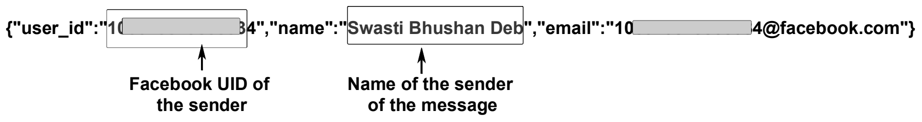
5.2 Analysis of Database Files

Analysis of the database files (see Figure 1 for the database files and their locations) revealed that they store a substantial amount of valuable user data. SQLite Manager was used to analyze the database files and

Table 6. Important databases and tables.

Database (.SQLite)	Table	Remarks
`Friends`	"friends"	Friends list of the Facebook test subject
`FriendRequests`	"friend_requests"	Friend requests sent to the test subject by Facebook users
`Messages`	"messages" "threads" "users"	Messages sent and received
`Stories`	"places"	Geographical locations in stories, updates, etc.
	"stories"	Stories in news feeds
`Notifications`	"notifications"	Notifications in a news feed when other Facebook users comment, like or share something

reconstruct the chat logs, friends list, notifications etc. Table 6 lists the important tables associated with the databases.



```
{"user_id":"1[redacted]4","name":"Swasti Bhushan Deb","email":"10[redacted]4@facebook.com"}
```

Figure 5. JSON-formatted "sender" column value.

Analysis of Facebook Chat Artifacts. Chat and message histories are threaded together [7]. Table 7 presents the tables in the `Messages` database and their forensic artifacts. As detailed in Figure 5 and Tables 6 and 7, the "messages" table of the `Messages.sqlite` database was found to contain forensically-important user data. The "content" column contains the textual content of chat communications and, along with the "sender" column data, can be used to determine who sent the messages.

The data in the "sender" column is in the JavaScript object notation (JSON) format. Figure 5 shows a typical "sender" column value.

The files in `C:\Users\{UserName}\AppData\Local\Packages\Facebook.Facebook_8xx8rvfyw5nnt\LocalState\UID\MessageAttachmentCache` contain the files and photos sent as attachments during a chat. These attachments are located in individual folders named after the MD5 hash values of the uploaded files with the same extensions as the files

Table 7. Recovered artifacts.

Table	**Recovered Artifacts**
"friends"	Facebook UID, registered email id, mobile numbers, names of friends Friends who used the Facebook Messenger app Communication ranks indicating the frequencies of interactions between the test subject and friends (likes, share, comments); close friends have communication ranks close to 1 (e.g., 0.75)
"friend_requests"	Names and UIDs of Facebook users who sent friend requests Local times when the friend requests were sent by Facebook users
"messages"	Attributes of messages communicated with friends such as individual/group chat sessions in plaintext, attachments in chat/message sessions, etc. Timestamps in the Unix epoch format when messages were sent or received by the test subject
"threads"	Unique thread id generated for each chat session; chats with unique users have unique thread ids Names, participants of chat groups, timestamp of the last chat communication in a message thread
"stories"	Public URLs of stories in news feeds; attachments available with stories along with their timestamps Status updates, tagged geolocations, friends and the privacy scope of posts; comments/likes and posts shared
"places"	Geolocations, if any, tagged in stories, along with the names of the geographical locations in plaintext Public URLs of pages located near the geographical locations
"notifications"	Facebook notifications in a news feed at a particular instant when other Facebook users comment, like or share Public URL of the notification, local time when the notification was created and whether or not the particular notification was read by the test subject

and photos. Deleting the attachments from a chat thread removes the files and images, but the parent folders are retained.

Analysis of Facebook Posts/Status Update Artifacts. A status/story update in the news feed of a user is reflected in the "stories" table of the `Stories.sqlite` database (Table 8). Traces of the test subject's post/comment on a friend's timeline are not directly recorded in a database. However, if a friend likes/comments/shares the post, they

Table 8. Important columns in the `Stories.sqlite` database.

Column Name	Forensic Significance
"creation_time"	Post/status update timestamp in the Unix epoch format
"url"	URL of the post/status update
"edit_history"	Number of times the post/status update was edited
"privacy_scope"	Privacy settings of a post/status update
"can_viewer_edit"	Value of one indicates post/status update by the test subject that cannot be edited by friends; value of zero indicates post/status update by a friend
"can_viewer_delete"	Value of one indicates post/status update by the test subject that cannot be deleted by friends; value of zero indicates post/status update by a friend
"message"	Indicates the comments, status update, etc.

are reflected in the "notifications" table of the `Notifications.sqlite` database. The "title_text" column values indicate the notifications that appear in the news feed.

Status updates by the test subject are stored as "can_viewer_delete" and "can_viewer_edit" column values in the "stories" table. A value of one indicates a status update by the test subject. A value of zero indicates an update by a Facebook friend.

The "edit_history" column values indicate changes/edits to posts. A value of zero indicates no changes, a value of one indicates that the post was edited once, and so on.

The "privacy_scope" column values detail the privacy settings of the account. The "privacy_scope" values may be queried by a forensic investigator who is interested in the status of the posts.

Analysis of Photo Upload Artifacts. An uploaded photo may include a caption. A photo may be uploaded from the locally-available photos or via the camera app interface. Photos uploaded in a post/photo album are located at `C:\Users\{UserName}\AppData\Local\Packages\Facebook.Facebook_8xx8rvfyw5nnt\TempState`. The date of creation of a photo (`.jpg file`) in this location corresponds to the time at which it was uploaded. Table 9 presents the naming conventions of the files found in the `TempState` directory. Intentionally deleting files in the directory does not delete their entries in the `Messages.sqlite` database.

Table 10 shows the registry keys that hold information about the most recently uploaded photos from different sources.

Analysis revealed that the "stories" table of the `Stories.sqlite` database records only the caption accompanying a photo. The "mes-

Table 9. Photo upload artifacts in the `TempState` directory.

Photos	**Naming Convention**	**Artifacts Found**
Locally-available photos	`x.jpg`, where `x` is the MD5 hash value of an uploaded photo	The actual uploaded photos with timestamps and the `.jpg` files named after the MD5 hash values of the photos
Facebook app camera photos	`picture00x.jpg`, where `x` is incremented for every new photo uploaded using the built-in camera interface	The actual uploaded photos with timestamps

Table 10. Registry keys with information about uploaded photos.

Registry Subkey	**Significance**
`KEY_USERS\SID\Software\Classes\LocalSettings\Software\Microsoft\Windows\CurrentVersion\AppModel\SystemAppData\Facebook.Facebook_8xx8rvfyw5nnt\PersistedStorageItemTable\System\688c235d-95bf-4ee0-af07-6b1058b568e9.Request.0`	Last uploaded photos; uploaded from the locally-available photos
`HKEY_USERS\SID\Software\Classes\LocalSettings\Software\Microsoft\Windows\CurrentVersion\AppModel\SystemAppData\Facebook.Facebook_8xx8rvfyw5nnt\PersistedStorageItemTable\System\c19f7613-f825-47a9-b5b9-5f44847eabc8.Request.0`	Last uploaded photos; uploaded via the camera interface of the app

sage" column reveals the caption if the "can_viewer_delete" and "can_viewer_edit" column values are one. The contents of the `TempState` directory and the "message" column of the `Stories.sqlite` database reveal correlations between the textual content and the uploaded photos. The creation dates of the `.jpg` files in the `TempState` directory are equivalent to the "creation_time" values that are adequate for reconstructing photo posts.

5.3 Analysis of Twitter Metro App Artifacts

This section presents information about Twitter metro app artifacts and their locations. Table 11 presents the contents of the `AC` directory and their forensic significance. Table 12 presents the contents of the `LocalState\TwitterID` directory and their forensic significance.

Twitter.sqlite Database. This database, which is located at `C:\Users\{Username}\AppData\Local\Microsoft\Windows\9E2F88E3.Twit`

Table 11. Forensic relevance of files in `AC`.

Directory/Files	**Forensic Relevance**
`INetCache`	Contains four randomly-numbered hidden folders; each folder contains files that encompass the default display pictures of Twitter followers
`INetCoookies`	Cookie information related to Facebook profiles
`INetHistory`	Empty

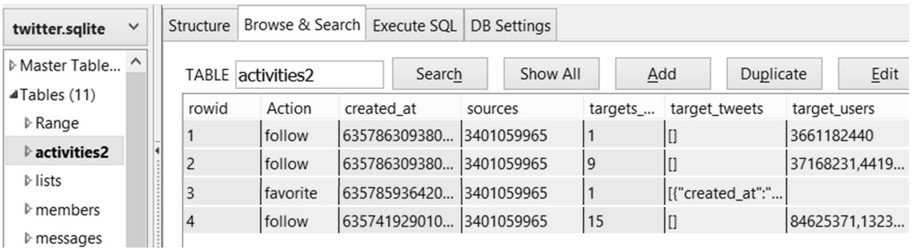
Table 12. Forensic relevance of files in `LocalState`.

Directory/Files	**Forensic Relevance**
`.local_cache`/Avatar	Contains randomly-numbered files corresponding to the avatar or header photos of the profiles of users followed by the test subject on Twitter
`Media`	Contains randomly-numbered files corresponding to the photos included in the tweets of users followed by the test subject
`Twitter.sqlite`	Contains information about latest tweets, private messages sent to followers, favorite lists, etc.

Table 13. Artifacts found in the `Twitter.sqlite` database.

Table	**Artifacts Found**
"activities2"	Information about actions such as follow and favorite, and the related timestamps
	Information about the users on whom the actions were performed
"messages"	Message content in plaintext, sender screen names and Twitter UIDs
"search_queries"	Search queries issued by the test subject
"statuses"	Contents of tweets made by the test subject and tweets of users followed by the test subject
	Tweet timestamps in the Unix epoch format, public URLs of tweets and geolocations
	Twitter UIDs of the authors of tweets, retweets, counts of retweets for each tweet
"users"	Screen names, locations, follower and friend counts; short URLs leading to tweets

`ter_wgeqdkkx372wm`, contains forensically-interesting information about the test subject. Table 13 lists the important tables in the database.

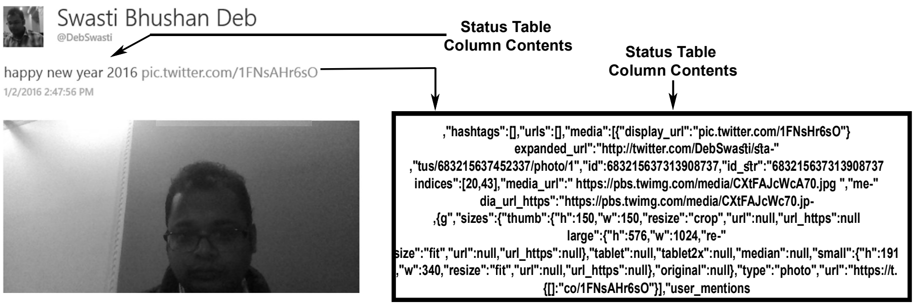


rowid	Action	created_at	sources	targets_...	target_tweets	target_users
1	follow	635786309380...	3401059965	1	[]	3661182440
2	follow	635786309380...	3401059965	9	[]	37168231,4419...
3	favorite	635785936420...	3401059965	1	[{"created_at":"...	
4	follow	635741929010...	3401059965	15	[]	84625371,1323...

Figure 6. The "activities2" table viewed using SQLite Manager.

Tweet Artifacts. The contents of a tweet may include captions, photos, etc. Photos may be uploaded from existing photos residing on the hard drive or via the built-in camera interface of the app. For each tweet, user-specific artifacts are found in the "statuses" table. Tweet captions are stored as "content" column values and tweet timestamps as "created_at" column values. The "entities" column values provide the contents of tweets posted by the test subject.

Figure 6 shows the forensically-important data that resides in the columns of the "activities2" in the `Twitter.sqlite` database. Note that the "activities2" table is viewed using SQLite Manager.



happy new year 2016 pic.twitter.com/1FNsAHr6sO
1/2/2016 2:47:56 PM

```
,"hashtags":[],"urls":[],"media":[{"display_url":"pic.twitter.com/1FNsHr6sO"}
expanded_url":"http://twitter.com/DebSwasti/sta-"
,"tus/683215637452337/photo/1","id":683215637313908737,"id_str":"683215637313908737
indices":[20,43],"media_url":" https://pbs.twimg.com/media/CXtFAJcWcA70.jpg ","me-"
dia_url_https":"https://pbs.twimg.com/media/CXtFAJcWc70.jp-
,{g","sizes":{"thumb":{"h":150,"w":150,"resize":"crop","url":null,"url_https":null
large":{"h":576,"w":1024,"re-"
size":"fit","url":null,"url_https":null},"tablet":null,"tablet2x":null,"median":null,"small":{"h":191
,"w":340,"resize":"fit","url":null,"url_https":null},"original":null},"type":"photo","url":"https://t.
{[]:"co/1FNsAHr6sO"}],"user_mentions
```

Figure 7. Reconstruction of a tweet using information from the "statuses" table.

For tweets involving photos, the photos taken via the camera interface are stored in the `TempState` directory, with the naming convention: `picture00x.jpg`, where `x` is incremented for every new photo uploaded using the camera interface. The creation timestamps of the `.jpg` files in the directory are the same as the corresponding "created" column values in the "statuses" table. Figure 7 shows the reconstruction of a tweet based on the "content" and "entities" columns of the "statuses" table.

Table 14. Summary of Facebook metro app artifacts.

Facebook User Activity	**Artifacts Found**
Private Messages	Messages in plaintext, file attachments, photo attachments (existing and those taken by the camera), sender UIDs, names, coordinates, timestamps when messages were sent/received
Notifications	Text included in notifications, URLs of posts, likes and shares by friends, read/unread status
Friends List	Names, UIDs, contact emails, phone numbers of friends, friends using messenger, communication rank, friend requests by other friends
Status Updates	Captions, privacy scope of posts, status update timestamps, coordinates, edit histories, photos uploaded, last photos uploaded, photos uploaded from locally-available photos or captured via the camera interface, status updates from friends
Friend/Page Searches	Tentative indications leading to searches, dates and times of the searches
Photo Albums	Names of albums, photos in albums, uploaded times, types of photos uploaded (locally available or taken via the camera interface), URLs in the albums, privacy scopes

Table 15. Summary of Twitter metro app artifacts.

Twitter User Activity	**Artifacts Found**
Private Messages	Messages in plaintext, sent/received timestamps, sender/receiver screen names and Twitter UIDs
Tweets	Posted tweets, captions and photos, timestamps, tweets by followers, favorite tweets, retweets
Search Queries	Search queries in plaintext

6. Discussion

The results presented in the preceding sections demonstrate that a wide range of artifacts related to Facebook and Twitter metro apps can be located and analyzed in digital forensic investigations. This section summarizes the results and discusses the main findings.

Tables 14 and 15 summarize the Facebook and Twitter metro app artifacts that were found in the experiments. The artifacts closely model the activities performed by the test subject.

A user who desires to hide his/her activities would typically edit posts or delete posts and messages. As discussed in Section 5, the majority of artifacts were found in SQLite databases. Thus, the modified and deleted activities can be reconstructed by recovering the corresponding SQLite records from the associated databases. The recovery of records and their reconstruction are facilitated by forensic tools such as Oxygen Forensics SQLite Viewer and Sanderson Forensics SQLite Recovery.

From a forensic standpoint, the volume shadow service (VSS) plays a significant role in reconstructing modified or deleted user activities because it creates and maintains multiple historical snapshots of the volumes on a disk. This service is advantageous when important files have been modified, rolled over or intentionally removed or deleted. Forensic examiners can cross-check shadow copies to (at the very least) determine if information was removed. The volume shadow service maintains a record of every block of data that has changed and only backs up a block if it is about to be modified; this enables it to store considerable data in a small amount of space.

The volume shadow service is available in Windows 8.x, but it is not accessible via Windows Explorer. The volume shadow copy service administrative command-line tool `vssadmin` may be used to display the list of volume shadow copy backups and all the installed shadow copy writers and providers in the command window. GUI-based tools such as ShadowExplorer may be used to access and display shadow copies.

7. Conclusions

Metro apps in Windows 8.x personal computer environments enable social media platforms to be accessed without an Internet browser. Since these apps leave valuable evidentiary traces of user activity, it is important to locate, analyze and reconstruct the traces in digital forensic investigations.

Unfortunately, little digital forensic research has focused on locating and analyzing Facebook and Twitter metro app artifacts in Windows 8.x personal computer environments. Indeed, the vast majority of studies have been limited to the analysis of social media apps on smartphones. The research described in this chapter involved an exhaustive analysis of the artifacts that remain after Facebook and Twitter metro apps are used. Digital forensic professionals should be aware that metro apps are a gold mine in investigations and, as the experimental results presented in this chapter reveal, activities piloted via metro apps can be located, identified and reconstructed, even if attempts have been made to modify or delete posts and tweets.

References

[1] N. Al Mutawa, I. Al Awadhi, I. Baggili and A. Marrington, Forensic artifacts of Facebook's instant messaging service, *Proceedings of the International Conference on Internet Technology and Secured Transactions*, pp. 771–776, 2011.

[2] N. Al Mutawa, I. Baggili and A. Marrington, Forensic analysis of social networking applications on mobile devices, *Digital Investigation*, vol. 9(S), pp. S24–S33, 2012.

[3] M. Brewer, T. Fenger, R. Boggs and C. Vance, A Comparison Between the Windows 8 and Windows 7 Registries, Forensic Science Center, Marshall University, Huntington, West Virginia (`www.marshall.edu/forensics/files/Matts-Paper.pdf`), 2014.

[4] M. Dickson, An examination into AOL Instant Messenger 5.5 contact identification, *Digital Investigation*, vol. 3(4), pp. 227–237, 2006.

[5] M. Dickson, An examination into Yahoo Messenger 7.0 contact identification, *Digital Investigation*, vol. 3(3), pp. 159–165, 2006.

[6] Facebook, Law Enforcement Online Requests, Menlo Park, California (`www.facebook.com/records/x/login`), 2016.

[7] Facebook Help Center, How does chat work with messages? Facebook, Menlo Park, California (`www.facebook.com/help/124629310950859`), 2016.

[8] T. Goh, Challenges in Windows 8 Operating System for Digital Forensic Investigations, M.F.I.T. Thesis, School of Computing and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand, 2014.

[9] A. Iqbal, H. Al Obaidli, A. Marrington and A. Jones, Windows Surface RT tablet forensics, *Digital Investigation*, vol. 11(S1), pp. S87–S93, 2014.

[10] Y. Khatri, Search History on Windows 8 and 8.1, Yogesh Khatri's Forensic Blog, Swift Forensics (`www.swiftforensics.com/2014/04/search-history-on-windows-8-and-81.html`), April 1, 2014.

[11] M. Kruzeniski, Welcome Twitter for Windows 8, Twitter, San Francisco, California (`blog.twitter.com/2013/welcome-twitter-for-windows-8`), March 14, 2013.

[12] C. Lee and M. Chung, Digital forensic analysis on Windows 8 style UI instant messenger applications, in *Computer Science and its Applications: Ubiquitous Information Technologies*, J. Park, I. Stojmenovic, H. Jeong and G. Yi (Eds.), Springer, Berlin Heidelberg, Germany, pp. 1037–1042, 2015.

[13] Microsoft Windows Dev Center, App and User Data, Microsoft, Redmond, Washington (`msdn.microsoft.com/en-us/library/windows/apps/jj553522.aspx`), 2016.

[14] Microsoft Windows Dev Center, App Data Storage, Microsoft, Redmond, Washington (`msdn.microsoft.com/en-us/library/windows/apps/hh464917.aspx`), 2016.

[15] C. Murphy, A. Leong, M. Gaffney, S. Punja, J. Gibb and B. McGarry, Windows Phone 8 Forensic Artifacts, InfoSec Reading Room, SANS Institute, Bethesda, Maryland, 2015.

[16] A. Parsons, Windows 10 Forensics Part 2: Facebook App Forensics, Computer and Digital Forensics Blog, Senator Patrick Leahy Center for Digital Investigation, Champlain College, Burlington, Vermont (`computerforensicsblog.champlain.edu/2015/04/01/windows-10-facebook-forensics`), April 1, 2015.

[17] J. Stormo, Analysis of Windows 8 Registry Artifacts, M.S. Thesis, Department of Computer Science, University of New Orleans, New Orleans, Louisiana, 2013.

[18] A. Thomson, Windows 8 Forensic Guide, M.F.S. Thesis, Department of Forensic Sciences, George Washington University, Washington, DC (`www.propellerheadforensics.files.wordpress.com/2012/05/thomson_windows-8-forensic-guide2.pdf`), 2012.

[19] Twitter, Law Enforcement Request, San Francisco, California (`support.twitter.com/forms/lawenforcement`), 2016.

[20] F. Zellers, MySpace.com Forensic Artifacts Keyword Searches (`www.inlanddirect.com/CEIC-2008.pdf`), 2008.